L Number	Hits	Search Text	DB	Time stamp
1	1434	concurrent near2 process	USPAT	2003/05/23 08:05
2	1441	713/?.ccls. or 380/?.ccls.	USPAT	2003/05/23 07:50
3	6	(concurrent near2 process) and (713/?.ccls. or 380/?.ccls.)	USPAT	2003/05/23 07:51
4	8993	(encrypt$3 or encipher or cipher$3)near2 (data or file or information or signal or program or software)	USPAT	2003/05/23 07:53
5	481	(while or concurrent) same (authenticat$3 or validat$3 or verify or verification or vaerfies)	USPAT	2003/05/23 07:54
6	7	((encrypt$3 or encipher or cipher$3)near2 (data or file or information or signal or program or software)) same ((while or concurrent) same (authenticat$3 or validat$3 or verify or verification or vaerfies))	USPAT	2003/05/23 07:58
7	2	(concurrent near2 process) same (authneticat$3 or encrypt$3)	USPAT	2003/05/23 07:59
8	0	(concurrent near2 process) same authenticat$3	USPAT	2003/05/23 08:06
9	72	(concurrent near2 process) and authenticat$3	USPAT	2003/05/23 09:19
10	5220	second near2 packet	USPAT	2003/05/23 09:21
11	2	(concurrent near2 process) same (second near2 packet)	USPAT	2003/05/23 09:20
12	0	(second near2 packet) same authneticat$3	USPAT	2003/05/23 09:22
13	25	(second near2 packet) same authenticat$3	USPAT	2003/05/23 09:23
14	5	((second near2 packet) same authenticat$3) and concurrent$3	USPAT	2003/05/23 09:25
15	49233	concurrent or parallel same (authenticat$3 and encrypt$3)	USPAT	2003/05/23 09:25
16	5022	authenticat$3 and encrypt$3	USPAT	2003/05/23 09:26
17	1209793	parallel or concurrent or concurrently	USPAT	2003/05/23 09:27
18	1255	first adj1 packet and second adj1 packet	USPAT	2003/05/23 09:31
19	0	(((encrypt$3 or encipher or cipher$3)near2 (data or file or information or signal or program or software)) same ((while or concurrent) same (authenticat$3 or validat$3 or verify or verification or vaerfies))) and ((concurrent near2 process) same (authneticat$3 or encrypt$3))	USPAT	2003/05/23 09:30
20	349	(first adj1 packet and second adj1 packet) and (concurrently or concurrent)	USPAT	2003/05/23 09:32
21	1	(713/?.ccls. or 380/?.ccls.) and ((first adj1 packet and second adj1 packet) and (concurrently or concurrent))	USPAT	2003/05/23 09:32
22	4	authenticat$3 near2 second adj1 packet	USPAT	2003/05/23 09:33
23	7	((while or concurrent) same (authenticat$3 or validat$3 or verify or verification or vaerfies)) same packet	USPAT	2003/05/23 09:40
24	40	(encrypt or cipher or encipher) same (authenticat$3 or validat$3 or verification or verfies or verifying) same (parallel or concurrently or concurrent or while)	USPAT	2003/05/23 09:47
25	26	process near2 packets near4 (simultaneously or concurrent or concurrently)	USPAT	2003/05/23 09:57

```
s (encrypt? or cipher? or encipher?)(3a)(second)(1a)(data or file? or packet?
or program? or information or key or password or ID)

          9909 ENCRYPT?
          2183 CIPHER?
          1118 ENCIPHER?
       1910950 SECOND
        629630 DATA
        872917 FILE?
         37550 PACKET?
        344318 PROGRAM?
        503093 INFORMATION
        208267 KEY
          7472 PASSWORD
         75970 ID
L1          720 (ENCRYPT? OR CIPHER? OR ENCIPHER?)(3A)(SECOND)(1A)(DATA OR
FILE?
                OR PACKET? OR PROGRAM? OR INFORMATION OR KEY OR PASSWORD OR
ID)

=> s (same time or while)(p)(authenticat? or verif?)(p)(first)(1a)(packet? or
information or program or data or file? or key or password or ID)

       2092012 SAME
       1809008 TIME
        538484 SAME TIME
                  (SAME(W)TIME)
             0 WHILE
          6766 AUTHENTICAT?
         91661 VERIF?
       2077545 FIRST
         37550 PACKET?
        503093 INFORMATION
        219174 PROGRAM
        629630 DATA
        872917 FILE?
        208267 KEY
          7472 PASSWORD
         75970 ID
L2           22 (SAME TIME OR WHILE)(P)(AUTHENTICAT? OR
VERIF?)(P)(FIRST)(1A)(PA
               CKET? OR INFORMATION OR PROGRAM OR DATA OR FILE? OR KEY OR
PASSW
               ORD OR ID)

=> s l1(p)l2

L3           0 L1(P)L2

=> d l2 1 kwic


L2   ANSWER 1 OF 22  USPATFULL
SUMM   An example illustrating the operation of the Boundary-Scan architecture
       will now be described. To verify the connection between first
       IC 110 and second IC 120 on copper trace 108, test control circuitry
       148(1) receives instructions to shift a first data
       value into a first data register 142(1) on first IC
       110. At the same time, test control circuitry 148(2)
```

on second IC 120 is instructed to shift a second data value into a second data register 146(2). Test control circuits 148(1) and 148(2) are then controlled to transmit the data value from **first data** register 146(1) and receive the data value in second data register 146(2) on trace 108. The data value stored in second data register 146(2) is then shifted out from second IC 120. If the shifted-out data value equals the **first data** value, then the connection between first IC 110 and second IC 120 on trace 108 is **verified**.

=> d l2 2 kwic

L2 ANSWER 2 OF 22 USPATFULL
DETD . . . 36 to the frame quality checking and lookup subsystem 38. With respect to subsystem 38, frame quality checking includes the **verification** of the frame version (CCSDS specifies version 1 and version 2 frames), frame sequence counts and fame length (compared to. . . . the frame synchronizer, and embedded in the frames. This subsystem is capable of processing two different frame versions at the **same time**. The lookup function refers to the ability to form addresses for accessing an external memory device. After capturing frame header **information**, the **first** two bytes of information which contain frame version, spacecraft ID and virtual channel ID are used to form an address.. . .

=> d l2 3 kwic

L2 ANSWER 3 OF 22 USPATFULL
DETD Thereafter, the **verification** operation is started by reading out word data in serial from the group of memory cells at the designated location in the memory cell array under the control of the controller 102. At the **same time**, the **first** latched **data** D.sub.L1 is read out in bits according to the bit address signal received from the controller 102.

=> d l2 4 kwic

L2 ANSWER 4 OF 22 USPATFULL
SUMM The memory medium which stores the server access **authentication** program according to the present invention registers the server address and the memory medium identification information. And also, the memory medium stores a **first program** to be used by a client computer. The computer use the **first program**, reading out such the server addresses and the memory medium identification information,and connecting to a desired server by using thus. . . In this case, the client uses thus read out server address, to be connected to the server and, at the **same time**, transmits thus read out the memory medium identification information to ask for server access permission.

=> d l2 5 kwic

L2 ANSWER 5 OF 22 USPATFULL

SUMM . . . M is known. Then when M has to be sent the signer computes a 1-time signature of M with the **authenticated** 1-time public key and sends out M tagged with the 1-time public key and the two signatures. Notice that the receiver must compute two signature **verifications**: one against the long-lived key to **authenticate** the 1-time key associated to M, and one against such 1-time key to **authenticate** M itself In our scheme we need to make both signing and **verification** extremely fast, and indeed in our case each block (except for the first) is signed (and hence **verified**) only once with a 1-time **key**. The **first** is signed and **verified** only once with a long-lived key. Old 1-time keys are used in order to **authenticate** new 1-time keys. This has appeared in several places but always for long-lived keys. Examples include [5,6,7] where this technique. . . by using the chaining technique with 1-time keys, embedding the keys inside the stream flow so that old keys can **authenticate** at the **same time** both the new keys and the current stream block.

=> d l2 6 kwic

L2 ANSWER 6 OF 22 USPATFULL
DETD **First**, magnetic **information** representing 50,000 yens, i.e., 500 units each of 100 yens, for renting balls to play a pinball game is recorded. . . 120 and 122 into contact with the opposite exposed extremities 115d of each conductive rubber line 115, respectively. At the **same time**, the electrodes 120a and 120b of the contact unit 120 are brought into contact with the exposed extremities 115d of. . . information recorded in the specified unit magnetic segment 12a and representing 500 units is the latest magnetic information through the **authentication** of the magnetic information with reference to the electrical information.

=> d l2 7 kwic

L2 ANSWER 7 OF 22 USPATFULL
DETD . . . the program memory 140 to call for the execution of the channel switching program stored in memory 140; at the **same time**, the parameters, including the desired output port and **authentication** identifier, are also passed to the channel switching program. It is further assumed that the channel switching program is a guarded program, so it will be necessary to execute a guard **program first**. The guard program in this example is one that compares the **authentication** identifier with a list of authorized **authentication** identifiers to ensure that the incoming **authentication** identifier is valid; the **authentication** list is stored, for example, in resource memory 150. Accordingly, the execution of the guard program is invoked initially (e.g., from the channel switching program) to test the **authentication** identifier.

=> d l2 8 kwic

L2 ANSWER 8 OF 22 USPATFULL
DETD Program **verification** will be explained. At **first**, **data** writing is effected. Namely, the word line driver 30 shown

in FIG. 6 selects the word line WLi. A select voltage is thereby applied
to the word line WLi. At the **same time**, the bit line selecting circuit 32 selects the select transistors 28v, 28n and 28n(i+1) connected the bit lines Blvi, Bli. . .

```
=> s (authenticat? or verif?)(p)(first)(1a)(packet? or information or program
or data or file? or key or password or ID)

          6766 AUTHENTICAT?
         91661 VERIF?
       2077545 FIRST
         37550 PACKET?
        503093 INFORMATION
        219174 PROGRAM
        629630 DATA
        872917 FILE?
        208267 KEY
          7472 PASSWORD
         75970 ID
L4        1816 (AUTHENTICAT? OR VERIF?)(P)(FIRST)(1A)(PACKET? OR INFORMATION
               OR PROGRAM OR DATA OR FILE? OR KEY OR PASSWORD OR ID)

=> s (encrypt? or cipher? or encipher?)(3a)(second)(1a)(data or file? or
packet? or program? or information or key or password or ID)

          9909 ENCRYPT?
          2183 CIPHER?
          1118 ENCIPHER?
       1910950 SECOND
        629630 DATA
        872917 FILE?
         37550 PACKET?
        344318 PROGRAM?
        503093 INFORMATION
        208267 KEY
          7472 PASSWORD
         75970 ID
L5         720 (ENCRYPT? OR CIPHER? OR ENCIPHER?)(3A)(SECOND)(1A)(DATA OR
FILE?
                OR PACKET? OR PROGRAM? OR INFORMATION OR KEY OR PASSWORD OR
ID)

=> s l4(p)l5

L6         117 L4(P)L5

=> s (same time or parallel or while)(p)l6

       2092012 SAME
       1809008 TIME
        538484 SAME TIME
                 (SAME(W)TIME)
        997980 PARALLEL
             0 WHILE
L7          11 (SAME TIME OR PARALLEL OR WHILE)(P)L6

=> d 17 1 kwic


L7   ANSWER 1 OF 11  USPATFULL


=> d 17 2 kwic
```

L7 ANSWER 2 OF 11 USPATFULL

=> d l7 3 kwic

L7 ANSWER 3 OF 11 USPATFULL

=> d l7 4 kwic

L7 ANSWER 4 OF 11 USPATFULL

=> d l7 5 kwic

L7 ANSWER 5 OF 11 USPATFULL

=> d l7 6 kwic

L7 ANSWER 6 OF 11 USPATFULL

=> d l7 7 kwic

L7 ANSWER 7 OF 11 USPATFULL

=> d his

(FILE 'HOME' ENTERED AT 07:50:16 ON 25 MAY 2001)

FILE 'USPATFULL' ENTERED AT 07:50:23 ON 25 MAY 2001

```
L1          720 S (ENCRYPT? OR CIPHER? OR ENCIPHER?)(3A)(SECOND)(1A)(DATA OR
FI
L2           22 S (SAME TIME OR WHILE)(P)(AUTHENTICAT? OR
VERIF?)(P)(FIRST)(1A)
L3            0 S L1(P)L2
L4         1816 S (AUTHENTICAT? OR VERIF?)(P)(FIRST)(1A)(PACKET? OR
INFORMATION
L5          720 S (ENCRYPT? OR CIPHER? OR ENCIPHER?)(3A)(SECOND)(1A)(DATA OR
FI
L6          117 S L4(P)L5
L7           11 S (SAME TIME OR PARALLEL OR WHILE)(P)L6
```